ISO TC 184/SC4 N1029
2000-04-20

ISO TC 184/SC4 STANDING DOCUMENT

Technical Committee 184 for Industrial Automation Systems and Integration
Subcommittee 4 for Industrial Data

Guidelines for the development of mapping specifications, 2nd edition

SC4 Secretariat
National Institute of Standards and Technology
Building 220/Room A127
Gaithersburg, Maryland 20899
USA

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

This standing document was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

International Standards produced by ISO/TC 184/SC4 are prepared according to guidelines put forth in the following standing documents:

— Guidelines for application interpreted construct development;

— Guidelines for application interpreted model development;

— Guidelines for the development and approval of STEP application protocols;

— Guidelines for the development of abstract test suites, 2nd edition;

— Guidelines for the development of mapping specifications, 2nd edition;

— ISO/TC 184/SC4 organization handbook;

— Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

The purpose of this standing document is to provide methods and procedures for the development of mapping specifications that appear within ISO 10303 application protocols. The mapping specification is a pivotal component of an application protocol (AP). The mapping specification documents the traceability of the application information requirements between the specification of these requirements in clause 4 of the AP and the application interpreted model (AIM) that documents how standardized constructs are applied to satisfy these requirements in clause 5 of the AP. This document is intended to provide guidance to application protocol development teams on the creation of mapping specifications. Additionally, this document may aid reviewers and implementors of APs in understanding mapping specifications. This document describes the methodology for producing a complete mapping specification, focusing on the development of the mapping specification content. Specifics on style, format, required text, and other presentation issues are provided in the *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition*. Additional guidance on other areas of AP development is found in the *Guidelines for development and approval of STEP application protocols*.

This is the second edition of this standing document. There are two primary changes from the first edition. First, the format of the mapping specification has been changed from a table to a nested subclause structure, and the term mapping specification has replaced the term mapping table. This has caused pervasive changes throughout the document, requiring removing all mention of tables or table components. Second, a syntax for defining one time and reusing portions of a reference path, called templates, is provided in this edition (see clause 4). Minor changes to the use of symbols in the reference path have also been made. The equal sign "=" may now be used to assign a specific type to an item of an aggregate (see 9.1.3). When using parenthesis, square brackets and angle brackets with complex paths where only a portion of the path starts at an integrated resource entity, that resource entity must now also be included in the reference path (see 9.1.4, 9.1.5, 9.1.6). Additionally, rules are no longer sequentially numbered and included in a list at the end of 5.1 of the AP. They are referenced through the subclause number of the subclause in the AP in which they are defined.

> NOTE - Issues logged against this document may be found in ISO TC 184/SC4 QC N135. Resolution of issues 9, 16, 17, 19, and 20 are incorporated in this edition. Resolution of 1, 2, 5, 12, 13, and 18 are left to future editions.

The mapping specification is organized by unit of functionality (UoF) as defined in clause 4.1 of the AP. Each information requirement of an AP is mapped to one or more constructs in the AIM. Each of these

mappings is documented in a subclause of the mapping specification for the UoF in which the application element is defined. Up to four types of information may be needed to completely specify the mapping of a given application element. These are: “AIM element”, “Source”, “Rules”, and “Reference path”. This document provides guidance on the how to populate each type of content in the mapping specification.

To aid in the understanding of this document, several examples taken from ISO 10303-201 are included within the body of this text. Additionally, a more comprehensive example is provided in annexes A and B of this document. In the example shown in the annexes, four application objects have been chosen from one of the UoFs in ISO 10303-201. Annex A contains the descriptions of the UoF, the application objects, and the application assertions as normatively documented in clause 4 of ISO 10303-201. Annex B contains the mapping specification for the application objects and assertions in annex A. Examples have been reformatted from the source AP to match the new clause structure for mapping specifications. Throughout the body of this text, references to application elements in examples are presented with leading capitals and underscores between the words. References to AIM elements in examples are presented in bold face with underscores between words, but no leading capitals.

Guidelines for the development of mapping specifications

1 Scope

This standing document specifies procedures and practices for the development and documentation of mapping specifications within ISO 10303 application protocols.

The following are within the scope of this standing document:

— description of the structure of mapping specifications for ISO 10303 application protocols;

— instructions for detailing the contents the mapping specification;

— description of the terms and syntax used within the mapping specification;

— explanation of when reference paths are required;

— procedures for identification of the start and end entities for required paths;

— examples that clarify the guidance provided within this document.

The following are outside the scope of this standing document:

— specification of font sizes and other layout information for the documentation of mapping specifications;

NOTE 1 - This information is found in *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition*.

— guidance on how to map the constructs of the ISO 10303 integrated resources to the information requirements of an ISO 10303 application protocol;

NOTE 2 - This information is available to a small extent in *Guidelines for application interpreted model development* and is planned to be covered in greater detail in a forthcoming document entitled *Guidelines for application interpretation*.

— guidelines for development of mapping specifications for documents other than ISO 10303 application protocols.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standing document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standing document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of the IEC and ISO maintain registers of currently valid International Standards.

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles.*

ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Associative draughting.*

The following documents contain provisions which, through reference in this text, constitute provisions of this standing document. At the time of adoption, the revisions of the documents indicated were valid. All documents are subject to revision, and users of this standing document are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below.

ISO/TC 184/SC4, *Guidelines for application interpreted model development*, N532, 1997

ISO/TC 184/SC4, *Guidelines for the development and approval of STEP application protocols*, N535, 1997

ISO/TC 184/SC4,[1] *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition.*

3 Terms, definitions and abbreviations

3.1 Terms defined in ISO 10303-1

For the purpose of this standing document, the following terms defined in ISO 10303-1 apply.

3.1.1
application
a group of one or more processes creating or using product data.

3.1.2
application context
the environment in which the integrated resources are interpreted to support the use of product data in a specific application.

[1]To be published.

3.1.3
application interpreted model (AIM)
an information model that uses the integrated resources necessary to satisfy the information requirements and constraints of an application reference model, within an application protocol.

3.1.4
application object
an atomic element of an application reference model that defines a unique concept of the application and contains attributes specifying the data elements of the object.

3.1.5
application protocol (AP)
a part of this International Standard that specifies an application interpreted model satisfying the scope and information requirements for a specific application.

> NOTE - This definition differs from the definition used in open system interconnection (OSI) standards. However, since this International Standard is not intended to be used directly with OSI communications, no confusion should arise.

3.1.6
application reference model (ARM)
an information model that describes the information requirements and constraints of a specific application context.

3.1.7
data
a representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers.

3.1.8
implementation method
a part of this International Standard that specifies a technique used by computer systems to exchange product data that is described using the EXPRESS data specification language [ISO 10303-11].

3.1.9
interpretation
the process of adapting a resource construct from the integrated resources to satisfy a requirement of an application protocol. This may involve the addition of restrictions on attributes, the addition of constraints, the addition of relationships among resource constructs**.**

3.1.10
product data
a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers.

3.1.11
resource construct
a collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of an aspect of product data.

3.1.12
unit of functionality (UoF)
a collection of application objects and their relationships that defines one or more concepts within the application context such that removal of any component would render the concepts incomplete or ambiguous.

3.2 Terms defined in ISO 10303-202

For the purpose of this standing document, the following terms defined in ISO 10303-202 apply.

3.2.1
application interpreted construct (AIC)
a logical grouping of interpreted constructs that supports a specific function for the usage of product data across multiple application contexts.

3.3 Abbreviations

For the purposes of this standing document, the following abbreviations apply.

AIC	application interpreted construct
AIM	application interpreted model
AP	application protocol
ARM	application reference model
UoF	unit of functionality

4 Mapping templates

A mapping specification may make use of mapping templates. A mapping template is conceptually similar to a programming language subroutine. Normally it consists of a signature, which may appear many times within a mapping specification, and a template body, which is documented once. Mapping templates that are predefined, such as the subtype and supertype reference templates described in 6.5 below, may not follow this structure completely; they may not have a template body. Mapping templates help to reduce the size of a mapping specification. In allowing for the reuse of both mappings and reference paths, mapping templates contribute to consistency and quality of a mapping specification. Mapping templates also facilitate maintenance of a mapping specification benefits from this concept.

The signature of a mapping template may appear in two areas of a mapping specification:

— in AIM element sub-clauses of application objects, attributes, or assertions (see 6.5);

— in reference paths (see 9.6).

The signature may use a static number of parameters as shown in example 1.

EXAMPLE 1 - /GROUPS(shape_aspect, 'boundary index 2')/

The non-blank characters following the first "/" define the name of the template. The name is written in uppercase. It shall not begin with a number; else numbers may be included. No special characters shall be used except for underscore ("_"). The name is followed by the list of parameters, which is surrounded by parentheses. The parameters are separated by commas. Parameters are written in lowercase. No additionalrestrictions apply to parameter names; they are either EXPRESS identifiers or strings in quotes.

The following notational conventions apply to mapping templates:

/: marks the beginning and the end of the signature of a mapping template;

( ): if within a set of "/", parentheses enclose one or several parameters;

,: if within a set of "/", comma separates two parameters and shall be followed by a single blank space;

': if within a set of "/", quotes enclose a text string;

&: introduces the name of a parameter within a template body.

Parameters that are not enclosed by quotes are names of EXPRESS data types.

The template bodies of mapping templates are documented in 5.1.1 of the APs that use this concept. See *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition* for details on the document structure and presentation style for mapping templates in APs.

A signature is the short-hand description of a template body. A signature can be replaced with the corresponding template body verbatim, except that its parameters need to be substituted by the values from the signature. A signature somewhere in a mapping specification may look like the one in example 1, above. The documentation of the corresponding template body in an AP is shown in example 2.

EXAMPLE 2 - A template body definition from 5.1.1 of an AP.

5.1.1.1 GROUPS

This mapping template assigns instances of an EXPRESS entity data type to an applied_group_assignment that performs a certain role. The description of that role is '.UNUSED.'

Signature:

/GROUPS(grouped_entity, 'arm_role')/

Parameter definition:

grouped_entity: the data type of the instances that are collected as grouped_items by an applied_group_assignment.

arm_role: the name of the object_role of the applied_group_assignment that collects the grouped_entitys.

Template body:

```
group_assignment.assigned_group
group_assignment =>
{group_assignment.role ->
object_role
[object_role.name = &arm_role]
[object_role.description = '.UNUSED.']}
applied_group_assignment
applied_group_assignment.items[i] ->
group_item
group_item = &grouped_entity
```

EXAMPLE - The following three lines show how the signature of this template may be embedded in a reference path:

```
group <-
/GROUPS(approval, 'approvals')/
approval
```

A mapping template may use other templates in its mapping body. The recursive use of templates is not foreseen. The appearance of mapping templates in mapping specifications is described in 6.5 and 9.6.

5 Application element

The application objects (i.e. entities and attributes) and assertions from clause 4 of the AP become subclause headings within the UoF mapping specification in accordance with the guidance provided in the *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition*. Each application element from the application protocol appears as the heading of at least one subclause of the specification. An application object may appear in more than one UoF within an AP. When this occurs, the subclause within each UoF mapping specification documents the mapping of that element within the context of that UoF.

The requirements of the application protocol may define more than one assertion between two application objects. When this occurs, each assertion is entered in a separate subclause of the mapping specification. The subclause heading is the heading of the application assertion from clause 4.3 of the AP along with an identifying phrase. The phrase is chosen from the normative text of the assertion and is placed in parentheses following the assertion heading. The assertions appear in the mapping specification in the order that they are defined in clause 4.3 of the AP.

EXAMPLE 1 - In this example, three assertions between a pair of application objects are documented in 4.3 of an AP and in the mapping specification. The mappings of the multiple assertions will appear as separate subclauses in the mapping specification. The details of the mapping are omitted here.

4.3.67 Structured_dimension_callout to Text_string

Each Structured_dimension_callout has as a dimension value one or more Text_string objects. Each Text_string may be the dimension value for exactly one Structured_dimension_callout.

Each Structured_dimension_callout has as a tolerance value zero, one, or many Text_string objects. Each Text_string may be the tolerance value for exactly one Structured_dimension_-callout.

Each Structured_dimension_callout has as unit text zero, one, or many Text_string objects. Each Text_string may be the unit text for exactly one Structured_dimension_callout.

5.1.2.1 STRUCTURED_DIMENSION_CALLOUT

<mapping of application object omitted>

5.1.2.1.1 structured_dimension_callout to text_string (as dimension value)

<mapping of assertion omitted>

5.1.2.1.2 structured_dimension_callout to text_string (as tolerance value)

<mapping of assertion omitted>

5.1.2.1.3 structured_dimension_callout to text_string (as unit text)

<mapping of assertion omitted>

When an application object maps to different AIM objects in different contexts, often the mappings can be clearly documented only through the use of numbered alternatives within the mapping specification. In these cases, the application element subclause contains numbered descriptions that indicate when the alternative mappings apply. The AIM elements and reference paths of the mapping specification are numbered correspondingly. In practice, when there are multiple mappings for an application object, these

mappings are frequently self-explanatory. However, mappings of assertions between objects with multiple mappings generally require the numbered descriptions.

EXAMPLE 2 - Below is an example of alternative mappings where clarification is provided.

5.1.x.y ANNOTATION_SUBFIGURE_DEFINITION_ELEMENT

#1: If the element is a curve, fill area, symbol, subfigure, or text
#2: If the element is a dimension or a draughting callout

AIM element:	#1: (draughting_annotation_occurrence)
	#2: (draughting_elements)
Source:	ISO 10303-201
	ISO 10303-201

In the example presented in annex B, the entity Organization has multiple mappings, though numbered alternatives are only provided for the mappings of the Organization_name attribute and the assertion from Approval to Organization.

NOTE - The cardinalities and inheritance documented in the information requirements of clause 4 of the AP are not visible in the mapping specification. The intent is that the inheritance in the ARM is preserved in the mapping, though the current mapping specification syntax provides no means to explicitly show this. For example, the mapping of an assertion from a supertype to another application element would apply also to subtypes of that supertype. Readers of the mapping specification must look to clause 4 of the AP for the cardinalities and subtype/supertype relationships among the application elements.

6 AIM element documentation

The AIM element contains the description of that to which the application element listed in the subclause heading. Application objects map to either a single AIM element, a combination of multiple AIM elements, or a choice among multiple AIM elements. AIM elements for application objects are entities or attributes from resource models (i.e., integrated resources or application interpreted constructs) or entities that are defined in the AIM of the AP being mapped. The AIM element for an application assertion is populated with the word “PATH” or the words “IDENTICAL MAPPING”. For both application objects and application assertions the AIM element may specify a close relationship to the mapping of another application element by using the mapping templates /SUBTYPE/ or /SUPERTYPE/. These mappings are described in the following subclauses.

The content of the AIM element is identified during the application interpretation process. This process is based on human understanding of the information requirements presented in the application protocol and the ISO 10303 integrated resources. This process is described in *Guidelines for application interpreted model development*.

The AIM element chosen for the mapping shall be as specific as possible. Mappings shall be shown to the attribute that will actually be populated with the data rather than to the entity itself. When the mapping

is to an entity, the entity name appears as the AIM element. When the mapping is to an attribute, the AIM element is the entity name where the attribute is defined, followed by a period, followed by the attribute name.

When a supertype is provided as the AIM element for a mapping, the intent is that the application object may map to the supertype itself or any of the subtypes that are within the scope of the AIM EXPRESS expanded listing as documented in annex A of the application protocol. If the intent is to map to a subset of the subtypes, the AIM element must be specified as a complex mapping (as described in 6.2) that indicates which subtypes are allowable. If the intent is to map the application object to the supertype only, the name of the supertype entity is surrounded by vertical bars "| |".

6.1 Single AIM element

A single AIM element is provided as the AIM element when only one entity or attribute from the integrated resources, application interpreted constructs, or AIM short form maps to the application object.

6.2 Multiple AIM elements

Sometimes a single application object maps to more than one AIM element. When this occurs, each mapping is documented in the mapping specification. This may occur as an "and" situation (both AIM elements must be present), as an "or" situation (either AIM element may be present), or as an "and/or" situation (one or more of the AIM elements must be present). The AIM elements provided in the mapping shall be as specific as possible. For instance, if the data that is to be populated is a combination of a unit and a value, it is more clear if the mapping is to both **measure_with_unit.unit_component** and **measure_with_unit.value_component** than simply to **measure_with_unit**.

In an "and" mapping, multiple AIM elements are required to satisfy the information requirement. Square brackets "[ ]" are placed around each required AIM element to denote the "and" situation. In an "or" situation, multiple AIM elements are alternatives that satisfy the information requirements. Parentheses "( )" are placed around each alternative AIM element to denote the "or" situation. In an "and/or" situation, at least one of the listed AIM elements is required to satisfy the information requirement. Angle brackets "< >" are placed around each alternative in the "and/or" situation. Parentheses, square and angle brackets may be used in combination in the specification of an AIM element to indicate the mapping of complex logical situations.

The alternatives in complex mappings (typically "or" situations) may be numbered to reflect descriptions provided for the application element (see clause 5). These descriptions are provided to clarify the situations under which the mappings apply.If the application of the different mappings is not clear, descriptions shall be provided. For an example of an attribute with multiple mappings, see the Organization_name attribute of the Organization application object in annex B. In this mapping, how the requirement for a name is satisfied depends on whether the organization is satisfied as a person, an organization, or a person within an organization and is thus indicated by parentheses. When a person within an organization is required, the name is satisfied by both the identification of the person and the name of the organization as indicated by the square brackets. The use of numbered alternatives is not mandated, but is left to the discretion of the AP team.

When an application element maps to the same AIM element but from different sources, the name of the AIM element is listed one time for each source. In the example below, the application object Point maps to the AIM entity **point** in the generic context defined by the integrated resource part and also in the context of an application interpreted construct. The entity name appears twice as the AIM element and the appropriate part numbers are referenced as the source.

EXAMPLE - In this example, the same entity is mapped to from two different sources.

5.1.x.y POINT

#1: If the point is not part of a shape_representation
#2: If the point is part of an elementary_brep_shape_representation

AIM element: #1: (point)
#2: (point)
Source: ISO 10303-42
ISO 10303-513

6.3 Path

The AIM element of a mapping specification is “PATH” when the application element is an application assertion and the application objects in the assertion map to different AIM elements. The reference path for an application assertion is designed to show how the relationship between the application objects is satisfied in the AIM.

There are cases where one of the application objects participating in an assertion maps to a complex logical relationship of AIM elements and the other maps to a subset of those AIM elements. Because the reader of the mapping specification cannot tell whether this was done on purpose or is an omission, when this case occurs, the AIM element for the assertion is still “PATH”, but the mapping shall be clarified with a note so it is clear that the reference path is not incomplete. The note shall follow the word “PATH”. The note shall be of the format:

NOTE 1 - For the purpose of this mapping, only the subset of the mapping of the <“to”_application_object> specified in the reference path is applicable.

Drawing_sheet to Sheet_placed_annotation assertion is an example of such an assertion (see example). The relationship between Drawing_sheet and Sheet_placed_annotation does not apply to case two where the annotation is a dimension or a draughting callout.

EXAMPLE - In this example a PATH AIM element requires additional clarification.

5.1.x.y DRAWING_SHEET

AIM element: dawing_sheet_revision

5.1.x.y.1 drawing_sheet to sheet_placed_annotation

AIM element: PATH

NOTE - For the purpose of this mapping, only the subset of the mapping of the Sheet_placed_annotation specified in the reference path is applicable.

Reference path: drawing_sheet_revision <=
presentation_area <=
presentation_representation <=
representation
representation.items[i] ->
representation_item =>
styled_item =>
annotation_occurrence =>
draughting_annotation_occurrence

5.1.x.z SHEET_PLACED_ANNOTATION

#1: If the annotation is a curve, fill area, symbol, subfigure, or text
#2: If the annotation is a dimension or a draughting callout

AIM element: #1: (draughting_annotation_occurrence)
#2: (draughting_elements)
Reference path: #1: (draughting_annotation_occurrence <=
annotation_occurrence)
#2: (draughting_elements <=
draughting_callout)

6.4 Identical mapping

The AIM element of a mapping contains the words “IDENTICAL MAPPING” when the application element is an application assertion and both application objects in the assertion map to the same AIM element instance. In the example below, Annotation_subfigure_definition_element maps to **draughting_annotation_occurrence** or to **draughting_elements**. Draughting_annotation maps to the same two choices. The assertion between these two application objects is an “IDENTICAL MAPPING” because these two application objects map to the same instance. Such cases may arise when there is a high level of detail in the application reference model (ARM) or the constructs in the integrated resources are more semantically rich than the constructs in the ARM.

EXAMPLE - The following is an example of an identical mapping.

5.1.x.y ANNOTATION_SUBFIGURE_DEFINITION_ELEMENT

#1: If the element is a curve, fill area, symbol, subfigure, or text
#2: If the element is a dimension or a draughting callout

AIM element: #1: (draughting_annotation_occurrence)
#2: (draughting_elements)

Source: ISO 10303-201
ISO 10303-201

5.1.x.y.1 annotation_subfigure_definition_element to draughting_annotation

AIM element: IDENTICAL MAPPING

5.1.x.z DRAUGHTING_ANNOTATION

#1: If the annotation is a curve, fill area, symbol, subfigure, or text
#2: If the annotation is a dimension or a draughting callout

AIM element: #1: (draughting_annotation_occurrence)
#2: (draughting_elements)
Source: ISO 10303-201
ISO 10303-201

When one of the application objects participating in an assertion maps to a complex logical relationship of AIM elements and the other maps to a subset of those AIM elements, the AIM element for the assertion is still IDENTICAL MAPPING, and the mapping may be clarified with a note stating that only the intersection of the AIM elements to which the two objects map are the same instance. The note clarifying the mapping, if it is deemed necessary, shall be included in as many places as it applies. The note shall be of the format:

NOTE - For the purpose of this mapping, only the intersection of the mappings for each of the application objects in the assertion is applicable.

6.5 Subtype and supertype cross references

When an ARM is specified as a data model that includes inheritance trees, it may happen that entities with an inheritance relationship map to the same AIM concepts.

NOTE 1 - The Quality Committee strongly discourages ARMs that are fully attributed data models.

Thus, mappings on lower levels of the inheritance trees may be identical to mappings higher up. Also, the mapping of supertypes may in some cases be merely the one or the sum of the mappings of its subtypes. To allow a mapping specification show these types of identical mappings the following two pre-defined mapping templates shall be used:

— /SUBTYPE(<application object that is subtype>)/ - see the mapping of this subtype of the current application object;

— /SUPERTYPE(<application object that is supertype>)/ - see the mapping of this supertype of the current application object.

The parameter of each of the templates shall be replaced by the name of the referenced application object.

NOTE 2 - The mapping templates /SUBTYPE/ and /SUPERTYPE/ are further documented in *Supplementary directives for the drafting and presentation of ISO 10303, 2nd edition*.

For both application objects, their attributes, and application assertions these templates may appear as the AIM element. For an application object the template replaces a common resource concept, such as an entity data type from the integrated resources. For an application assertion it replaces PATH or IDENTICAL MAPPING. If subtype or supertype cross referencing is applied, the remainder of the mapping specification that normally follows the AIM element sub-clause shall be omitted. The mapping specification documented under the referenced subtype or supertype is valid instead.

In addition to the subtype or supertype template, the AIM element shall also include a reference to the subclause containing the related AIM element in the mapping specification. Example 1, below, shows such a subclause reference. The AIM element for this application object is the same as for its subtype, i.e., in this example the application object chamfer. Chamfer is mapped in subclause 5.1.x.y, with x denoting the UoF and y the application object within the UoF.

EXAMPLE 1 - Sub-clause referencing among application objects:

5.1.2. Manufacturing_features UoF

5.1.2.4 Feature

AIM element: /SUBTYPE(chamfer)/5.1.x.y

One AIM element may contain several references to subtypes or supertypes. The "and", "or" and "and/or" operators of the mapping notation described in 9.1.3, 9.1.4, 9.1.5 shall be applied to indicate the relationships between the various mappings.

EXAMPLE 2 - An AIM element may map according to which subtype it is instantiated as.

AIM element: (/SUBTYPE(corner_cutout)/ 5.1.5.2)
(/SUBTYPE(interior_cutout)/ 5.1.7.6)

7 Source documentation

The source entry contains an ISO standard number and part number for each AIM element provided. In general,the part identifier that is the ISO standard in which the AIM element is defined (e.g. ISO 10404-41). The part numbers referenced in a mapping specification may correspond to an SC4 common resource, an integrated resource; an application interpreted construct; or the application protocol itself, in the case where the AIM element is an AP created specialization of an integrated resource entity. When an application object or assertion is mapped to an entity or type that is defined in the integrated resources, implicitly or explicitly brought into the scope of an AIC according to the interfacing rules of

EXPRESS as documented in ISO 10303-11, and the mapping is within the context of the AIC, the AIC part number is referenced as the source. If the mapping is not within the context of the AIC, the part number of the integrated resource construct is referenced as the source.

If the AIM element is either “PATH” or “IDENTICAL MAPPING”, no source document is listed.

8 Rules documentation

The mapping specification contains subclause references to the global rules in the AIM short form that constrain the mappings. The references correspond to the subclause number where the global rule is defined in the AP (e.g., 5.2.4.32). There may be more than one rule constraining a given mapping. Some mappings may be unconstrained. Rules restricting instantiation of entities within the AIM are to be included in the mapping specification only when the mapping is to an AIM element that shall not be independently instantiable. When an entity constrained by an instantiability rule appears in a reference path, that mapping assumes that the entity will be instantiated in the context of other entities; therefore, the reference to the rule is not needed for that mapping. All rules that are created in the AP short form, including entity instantiability rules, shall be referenced in the mapping specification at least once.

Each AP contains a global rule constraining **application_context**. This rule shall be referenced in any mappings to the following AIM elements: **product**, **product_context**, **product_definition**, and **library_assignment**.

9 Reference path documentation

The reference path illustrates how the requirements and relationships stated in clause 4 of the AP are maintained as a result of the application interpretation process. It specifies the complete path of entity references in the AIM that is needed to represent the information requirements of the ARM. A set of symbols and formats were developed to construct a consistent syntax for documenting reference paths. Reference path syntax is consistent for each type of application element. The intent of the current syntax is to facilitate human readability of the mapping specification. In the future, this syntax may be extended to improve computer readability of the mapping specification. This clause discusses the symbology used in documenting reference paths as well as reference path requirements for each type of application element.

9.1 Symbology

A reference path specification is read from left to right and from top to bottom. Each line of the reference path specification may contain symbols to illustrate the EXPRESS structure of the AIM objects. Understanding the symbology used in the reference path specification is the key to reading and writing mapping specifications.

9.1.1 Delimiter symbols

The delimiter symbols are used to indicate the relationship of the specified entity or attribute preceding the symbol to the specified entity or attribute following the symbol. The symbol should be placed at the end of the line, so that the name of the following entity or attribute is at the beginning of the next line. The delimiters should be separated from the entity or attribute text by a single space, for readability. The meaning of these symbols can be paraphrased as:

=> :"is a supertype of"
<= :"is a subtype of"
-> : "references"
<- : "is referenced by"

The "=>" and "<=" symbols indicate a supertype or subtype structure. The "=>" symbol is used to indicate that the specified entity preceding the symbol is the supertype of the entity specified on the next line. The "<=" symbol is used to indicate that the specified entity preceding the symbol is a subtype of the entity specified on the next line.

The "->" and "<-" symbols indicate the reference to an entity or type by an attribute. The "->" symbol is used to indicate that the specified attribute preceding the symbol references the entity or type specified on the next line. The "<-" symbol is used to indicate that the specified entity or type preceding the symbol is referenced by the attribute specified on the next line.

When an entity name appears on a line that is terminated without the use of one of the above delimiter symbols, this may indicate that the specified entity has the attribute shown on the next line. When an attribute name appears on a line that is terminated without the use of one of the above symbols, this may indicate that the specified attribute is an attribute of the entity shown on the next line. A new line is used without conveying additional semantics before and after lines where an EXPRESS select type value is provided (see 9.1.3). New lines precede and follow mapping rules (braces), and the individual options for mappings with multiple AIM elements (parentheses, square and angle brackets). A new line also terminates the reference path.

Reference path statements that are too long to fit on one line can be split using the forward slash symbol "\". The symbol conveys no additional meaning within the reference path. The "\" is positioned between elements of the statement, at the end of the line, preferably in white space. The "\" may appear between an entity and an attribute, following the period, but this case should be avoided where possible. The "\" should not be placed inside of a text string; the entire text string should follow the forward slash on the next line.

9.1.2 Aggregation symbols

If an attribute references an aggregate cardinality, and any single instance of the aggregate is of interest, brackets and the letter i "[i]" are used to indicate this. This reflects the usual requirement that the path can go through any member of the aggregate. The use of "[n]" (where n is an integer) indicates that member n of the aggregate is of interest in the mapping. A reference to an_entity.aggregate[1] reflects the requirement that the path must go through the first element of the aggregate. In order to limit the number of elements in an instantiation of the aggregation, EXPRESS rules shall be written in the short

form of the AIM. In the mapping of the Drawing to Approval assertion in annex B, the attribute **approved_items** is a set that references the select type **approved_item**.

9.1.3 Equal sign

An equal sign "=" is used in the reference path specification to indicate a member of the select list of an EXPRESS select type, an item from the enumerated list of an EXPRESS enumeration type, a specific value for an attribute, or a specific type is being assigned to an item of an aggregate. In the case of a select list, the name of the select type appears first followed on the same line by the equal sign and the member that is being selected. In the case of an enumerated list, the name of the enumeration type appears first followed on the same line by the equal sign and the enumerated item. In the case of a specific value, the attribute name appears first followed on the same line by the equal sign and the value assigned to the attribute. In the case where a specific type is being assigned to an item of an aggregate, the aggregate reference appears first followed on the same line by the equal sign and the specified item.

See the mapping of the Date attribute of the Approval object in annex B. In this reference path specification, the attribute **date_time** references the select type **date_time_select**. For this mapping, the selection is a **date**. As seen in this example, the name of the select type appears on the line before the line containing the equal sign, and the selected member appears on the line following the line containing the equal sign. The order of these lines is reversed for a reference path in which the selected member is referenced first in the path. See the mapping of the Drawing to Approval assertion in the example in annex B. In this mapping, the reference path encounters **drawing_revision** first, which is the selected member of the **approved_item** select type. In the following example, an item of an aggregate is being specified to be of a certain type; the **representation_item** must be a **measure_representation_item**.

EXAMPLE - The express defined types are used in a reference path below to illustrate assigning a specific type to an item of an aggregate.

```
ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
    item_element : compound_item_definition;
END_ENTITY;

TYPE compound_item_definition = SELECT
   (set_representation_item,
    list_representation_item);
END_TYPE;

TYPE set_representation_item = SET[1:?] OF representation_item;
END_TYPE;

TYPE list_representation_item = LIST[1:?] OF representation_item;
END_TYPE;

Reference path:   {compound_representation_item
                   compound_representation_item.item_element ->
                   compound_item_definition
                   compound_item_definition = list_representation_item
```

list_representation_item
list_representation_item[i] = representation_item
representation_item =>
measure_representation_item}

9.1.4 Parentheses

Parentheses "( )" are used to indicate the existence of options in the reference path. Each option is enclosed by a set of parentheses. The parentheses are used to indicate that a mapping has multiple reference paths or sections of the reference path. There are two reasons that the reference path may diverge: an object is mapped to multiple AIM entities or the reference path depends on the instantiation of the AIM. To aid understanding, the optional sections of the path may be numbered and a description provided, with the application object, that gives the reason for the divergence.

When an ARM object maps to an entity from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an "or" of an entity from the integrated resources and an entity defined in the AP, the entity from the integrated resources shall be enclosed in parenthesis in the reference path for clarity.

EXAMPLE - Fragments of a mapping specification that illustrate the "or" case.

AIM element:	(product_definition)
	(externally_defined_product_definition)
Reference path:	(product_definition)
	(externally_defined_product_definition <= [product_definition]
	[externally_defined_item])

See the mapping of the Approval to Organization assertion in annex B. In this example, it is necessary to show the reference paths for each of the AIM elements to which the Organization maps.

9.1.5 Square brackets

Square brackets "[ ]" are used to indicate two or more required sections of the reference path. The square brackets indicate that there are either multiple mappings or multiple paths required to satisfy the mapping. Each mapping or path is enclosed by a set of square brackets. To fully document how the requirements are satisfied by the mapping, sections of the path may be numbered and a description giving the reason for the divergence may be provided immediately following the subclause heading.

When an ARM object maps to an entity from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an "and" of an entity from the integrated resources and an entity defined in the AP, the entity from the integrated resources shall be enclosed in square brackets in the reference path for clarity.

See the mapping of the Annotation_subfigure_definition to 2D_cartesian_coordinate_space assertion in the example below. This example shows that every **representation_context** that satisfies the requirements of the application object **2D_cartesian_coordinate_space** must be both a **geometric_-representation_context** and a **global_unit_assigned_context**.

EXAMPLE - An example of a mapping that illustrate the “and” case.

5.1.x.y ANNOTATION_SUBFIGURE_DEFINITION

AIM element: symbol_representation_map
Source: ISO 10303-46
Reference path: {symbol_representation_map <=
representation_map
representation_map.mapped_representation ->
representation =>
symbol_representation =>
draughting_subfigure_representation}

5.1.x.y.1 annotation_subfigure_definition to 2d_cartesian_-coordinate_space

AIM element: PATH
Reference path: symbol_representation_map <=
representation_map
representation_map.mapped_representation ->
representation
{representation =>
symbol_representation =>
draughting_subfigure_representation}
representation.context_of_items ->
representation_context =>
[geometric_representation_context]
[global_unit_assigned_context]

5.1.x.z 2D_CARTESIAN_COORDINATE_SPACE

AIM element: [geometric_representation_context]
[global_unit_assigned_context]
Source: ISO 10303-42
ISO 10303-41

See the mapping of the Organization_name attribute of Organization in annex B. Where the mapping is to a person within an organization, both reference paths shown are required.

9.1.6 Angle brackets

Angle brackets “< >” are placed around elements of a reference path to indicate that at least one of the sections of the reference path enclosed by the angle brackets are required. The angle brackets indicate that there are either multiple mappings or multiple paths required to satisfy the mapping. To fully document how the requirements are satisfied by the mapping, sections of the path may be numbered and

a description giving the reason for the divergence may be provided immediately following the subclause heading.

When an ARM object maps to an entity from the integrated resources, no reference path is typically required. In the case where an ARM object maps to an "and/or" of an entity from the integrated resources and an entity defined in the AP, the entity from the integrated resources shall be enclosed in angle brackets in the reference path for clarity.

Mappings and reference paths that require the use of angle brackets are an indication that the application element being mapped represents multiple concepts. In these cases, the application element should be reviewed and the creation of a separate object for each concept should be considered.

9.1.7 Braces

Braces "{ }" are used to indicate constraints placed on the mapping. The constraints documented within the braces are commonly referred to as mapping rules. Mapping rules constrain the set of valid data populations. They apply only to the mapping of the application element being mapped in that subclause of the mapping specification. Because of this limited scope, it may not be possible to also write these constraints in the AIM EXPRESS as either global or local rules. Where possible, EXPRESS constraints should be provided in the AIM to document mapping rule constraints.

NOTE 1 - 9.2 of ISO 10303-11 enumerates the type of constraints that can be written for EXPRESS entities.

Mapping rules are used when the reference path specification would not normally contain AIM entities which are crucial to the mapping. Mapping rules may be used to include into the reference path specification required supertypes or subtypes, required values assigned to attributes, or AIC entities containing rules that constrain the mapping.

Mapping rules shall start on a new line of a reference path and shall end with a new line. The reference path shall be completely specified without the mapping rule; if the mapping rule were removed, the reference path would still be correct. If a reference path contains multiple mapping rules, all the rules apply to the mapping. If the granularity of the ARM and AIM differ such that the resulting mapping rules apply to different cases, provide numbered alternatives immediately following the subclause heading and number the mapping rules correspondingly in the reference path.

Mapping rules are also used when an assertion is mapped to a specialization of a resource entity and the reference path specification would not otherwise show the resource entity. The inclusion of these AIM elements is intended to satisfy the requirement that all mappings to a specialized entity shall have a reference path to a resource entity.

In this example, the reference path specification would not normally show the ISO 10303-201 created subtype. However, for this mapping, the only **draughting_callout_relationship** that satisfies the information requirement is the **dimension_callout_component_relationship** subtype. The second mapping rule indicates that the name of the relationship is restricted for this mapping to have the value "prefix".

EXAMPLE 1 - The mapping of the Structured_dimension_callout to Draughting_callout assertion contains two mapping rules.

5.1.x.y STRUCTURED_DIMENSION_CALLOUT

AIM element: structured_dimension_callout
Source: ISO 10303-201
Reference path: structured_dimension_callout <=
draughting_callout

5.1.x.y.1 structured_dimension_callout to draughting_callout (as prefix)

AIM element: PATH
Reference path: structured_dimension_callout <-
draughting_callout_relationship.relating_dimension_callout
draughting_callout_relationship
{draughting_callout_relationship <=
dimension_callout_component_relationship}
{draughting_callout_relationship.name = 'prefix'}
draughting_callout_relationship.related_draughting_callout ->
draughting_callout

5.1.x.z DRAUGHTING_CALLOUT

AIM element: draughting_callout
Source: ISO 10303-101

Another type of mapping rule is used to signify that any number of relationship entities may be assembled in a relationship tree structure. This is shown by the use of the relationship entity name followed by an asterisk "*". There should be white space between the entity name and the asterisk. An asterisk may only be used in a mapping rule, and the only currently known application of this syntax is in extending relationship trees to any number of levels. In the absence of a mapping rule including an asterisk, the path contains only the number of relationship entities shown. In the reference path in example 2, below, there may be any number of draughting_callout_relationship entities in the relationship tree.

EXAMPLE 2 - A mapping rule using the * operator.

draughting_callout <-
draughting_callout_relationship.related_draughting_callout
draughting_callout_relationship
{draughting_callout_relationship *}
draughting_callout_relationship.relating_draughting_callout ->
draughting_callout

NOTE 2 - There is an issue logged against this syntax element. No end condition is provided in the syntax, therefore this operator may not be automatable. This issue is under investigation. See ISO TC 184/SC4 QC N135, issue 18.

9.1.8 Vertical bars

Vertical bars "| |" are used to indicate that the mapping is only to the supertype entity found between the vertical bars. If the vertical bars are not present, and the entity being referenced is a supertype, it is assumed that any of the subtypes of that supertype are valid in the context of that reference path. The vertical bars limit the reference path to the supertype entity itself. Vertical bars may appear in both the AIM element and the reference path.

EXAMPLE - Use of vertical bar notation.

```
A.b ->
| only_supertype_allowed |
```

9.1.9 Single quotes

Single quotes " ' ' " are used to indicate that the enclosed text is intended to be the value of an attribute whose underlying type is a string. Single quotes are used in this context to provide consistency with ISO 10303-11. They may appear in the reference path only.

9.2 Application objects

A reference path specification is necessary for each application object that is mapped to a specialization of an integrated resource entity. The reference path starts with the AIM element to which the application object is mapped. It concludes at the integrated resource entity of which the specialization is a subtype. See the mapping of Drawing in annex B. In this example, Drawing is mapped to **draughting_drawing_revision**, which is a subtype of **drawing_revision** in ISO 10303-101. The reference path contains these two entities.

A reference path can also be shown to clarify a restriction on the mapping. See the mapping of Annotation_subfigure_definition in the example in 9.1.5. This object maps to an entity in the integrated resources; therefore, no reference path specification is required. However, to satisfy the requirements for this mapping, the inherited attribute **mapped_representation** must be of type **draughting_subfigure_representation**. This restriction is shown by including this portion of the reference path specification within braces (see 9.1.7).

In the case where a mapping has options and/or multiple required portions where one of the selections is an entity from the integrated resources, the entity from the integrated resources shall also appear in the reference path to completely state the options in the path. See the example in 9.1.4.

9.3 Application attributes

There are different mappings of an attribute that must be considered for the documentation of the reference path. An attribute may be mapped to an attribute of the same AIM entity to which the application object is mapped. In this case, no reference path is necessary for the attribute. See the mapping of the Description attribute of Approval in the example in annex B.

An attribute may be mapped to an attribute of an entity different from the one to which the application object is mapped. The reference path for the attribute starts with the AIM element to which the application object is mapped. The path follows the entities and attributes of the AIM to the AIM attribute to which the application attribute is mapped.

See the mapping of the Drawing_number attribute of Drawing in the example in annex B. In this example, Drawing is mapped to **draughting_drawing_revision** so the reference path of Drawing_-number begins with this entity. **Drawing_revision** has an attribute **drawing_identifier** that references the entity **drawing_definition**. The **drawing_definition** entity has an attribute **drawing_number** to which the Drawing_number attribute is being mapped.

9.4 Application assertions

Application assertions specify the relationships between pairs of application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects.

The reference path of an assertion starts with the AIM element to which the first application object is mapped. The path concludes at the AIM element to which the second application object in the assertion is mapped. See the mapping of the Drawing to Approval assertion in annex B. In this example, Drawing is mapped to **draughting_drawing_revision** so the reference path of the assertion begins with this entity. **Drawing_revision** is one of the **approved_items** to which **approval** is assigned.

If the AIM element is “IDENTICAL MAPPING”, no reference path specification is required.

In rare cases, an application assertion may map to an AIM entity or attribute. In these cases, the mapping may be to an attribute in the reference path that connects the two identified application objects, or to an AIM entity that acts as the intersection entity connecting the two identified application objects. This attribute or entity is selected as the AIM element with agreement by AIM interpretation experts. The source is the number of the part containing this entity or attribute.

9.5 AIC considerations

AICs define entities where the global rules pertaining to that AIC have been localized. These entities are called “root node” entities. If the root node of the AIC includes restrictions that apply to the mapping, the root node shall be included in the reference path specification to indicate this, even if the inclusion is only within a mapping rule.

9.6 Templates in reference paths

The reference paths of a mapping specification often repeat themselves. This is partly because of the limited size of the Integrated Resources. Also the existence of an inheritance tree in the ARM leads to repetition. The mapping templates that were introduced in clause 4 may be used to avoid such repetition. A piece of a reference path that appears several times within the mapping specification of an AP may be transformed into a mapping template. The bit of the reference path that shall be removed becomes the mapping body; this shall be documented in 5.1.1 of the AP. It will be replaced

in the reference path by the signature of the mapping template. Mapping body and signature may be designed to use parameters; this is recommended to increase the reusability of the mapping template.

The following examples build on previous ones and show reference paths without and with mapping templates:

EXAMPLE 1 - The reference path for the attribute boundary_index_2 (5.1.2.1.3) does not use mapping templates:

5.1.2 Manufacturing_features UoF

5.1.2.1 Corner_cutout_boundary_relationship

AIM element: group
Source: ISO 10303-41

5.1.2.1.3 boundary_index_2

AIM element: shape_aspect
Source: ISO 10303-41
Rules: 5.2.5.38
Reference path: group <-
group_assignment.assigned_group
group_assignment =>
{group_assignment.role ->
object_role
[object_role.name = 'boundary index 2']
[object_role.description = '.UNUSED.']}
applied_group_assignment
applied_group_assignment.items[i] ->
group_item
group_item = shape_aspect
shape_aspect

5.1.2.2 Feature

AIM element: /SUBTYPE(chamfer)/5.1.x.y.z

EXAMPLE 2 - An equivalent mapping specification to example 1, above, using the mapping template defined in example 2 of clause 4:

5.1.2 Manufacturing_features UoF

5.1.2.1 Corner_cutout_boundary_relationship

AIM element: group
Source: ISO 10303-41

5.1.2.1.3 boundary_index_2

AIM element: shape_aspect
Source: ISO 10303-41
Rules: 5.2.5.38
Reference path: group <-
/GROUPS(shape_aspect, 'boundary index 2')/
shape_aspect

5.1.2.2 Feature

AIM element: /SUBTYPE(chamfer)/5.1.x.y.z

Annex A

Example information requirements

This annex contains example descriptions of a UoF, application objects, and application assertions that appear in the example mapping specification in annex B. These descriptions are extracted from clause 4 of ISO 10303-201. Some objects, attributes, and assertions from the UoF have been excluded from the example.

NOTE - The numbering in this annex reflects the clause numbering as published in ISO 10303-201.

4.1 Units of functionality

4.1.4 drawing_structure_and_administration

The drawing_structure_and_administration UoF contains information about the hierarchical organization of drawings, drawing sheets, and drawing views, together with the administrative information necessary to manage drawings and drawing sheets. Drawing sheets and drawing views are defined in their specific coordinate space. Annotation may be assigned to each drawing sheet and drawing view. The administrative information supports the exchange of drawings between environments in which configuration management of drawings is used. The following application objects are used by drawing_-structure_and_administration UoF:

— Approval;

— Drawing;

— Drawing_sheet;

— Organization.

4.2 Application objects

4.2.13 Approval

An Approval is information that indicates a drawing, drawing sheet, or both have been reviewed for data content and for correctness of the presentation of that data and has been found to be acceptable. The data associated with an Approval are the following:

— Date;

— Description.

4.2.19.1 Date

The Date specifies the date on which the approval was assigned.

4.2.19.2 Description

The Description specifies the organization-specific release status or the authorized modifications for the revision of the drawing, drawing sheet, or both.

4.2.30 Drawing

A Drawing is the presentation of product data in a human-interpretable form wherein the physical and functional requirements for that product are presented pictorially and textually. The data associated with a Drawing are the following:

— Drawing_number;

— Drawing_revision_id.

4.2.30.2 Drawing_number

The Drawing_number specifies the identification of a particular drawing by an organization.

4.2.30.3 Drawing_revision_id

The Drawing_revision_id specifies the identification of a particular version of the drawing.

4.2.31 Drawing_sheet

A Drawing_sheet is a logical division of a drawing into a two-dimensional area for the presentation of product data. These divisions correspond to sheet paper sizes for plotting. A Drawing_sheet contains at least one Drawing_view or one Draughting_annotation. The data associated with a Drawing_sheet are the following:

— Sheet_number;

— Sheet_revision_id.

4.2.31.2 Sheet_number

The Sheet_number specifies the page number for a particular drawing sheet and its location in relation to other sheets of the drawing.

4.2.31.3 Sheet_revision_id

The Sheet_revision_id specifies the identification of a particular version of the drawing sheet.

4.2.57 Organization

An Organization is a number of persons or groups that designs, produces and supplies products and services. The data associated with an Organization are the following:

— Organization_name.

4.2.57.2 Organization_name

The Organization_name specifies the identification of a particular organization.

4.3 Application assertions

4.3.18 Approval to Organization

Each Approval is provided by one or more Organization objects. Each Organization provides zero, one, or many Approval objects.

4.3.32 Drawing to Approval

Each Drawing is governed by zero, one, or many Approval objects. Each Approval governs zero or one Drawing.

4.3.33 Drawing to Drawing_sheet

Each Drawing consists of one or more Drawing_sheet objects. Each Drawing_sheet belongs to exactly one Drawing.

4.3.37 Drawing_sheet to Approval

Each Drawing_sheet is governed by zero, one, or many Approval objects. Each Approval governs zero, one, or many Drawing_sheet objects.

Annex B

Example mapping specification

This annex contains the mapping specification that corresponds to the example information requirements in annex A. See annex A for the textual descriptions of the application objects.

The AIM entities found in this mapping specification are defined in ISO 10303-41 [4], ISO 10303-101 and ISO 10303-201.

5.1.4 Drawing_structure_and_administration UoF

5.1.4.1 APPROVAL

AIM element: approval
Source: ISO 10303-41

5.1.4.1.1 date

AIM element: calendar_date
Source: ISO 10303-41
Reference path: approval <-
approval_date_time.dated_approval
approval_date_time
approval_date_time.date_time ->
date_time_select
date_time_select = date
date =>
calendar_date

5.1.4.1.2 description

AIM element: approval.level
Source: ISO 10303-41

5.1.4.1.3 approval to organization

#1: If the approval is given by only a person
#2: If the approval is given by only an organization
#3: If the approval is given by a person within an organization

AIM element: PATH
Reference path: approval <-

approval_person_organization.authorized_approval
approval_person_organization
approval_person_organization.person_organization ->
person_organization_select
#1: (person_organization_select = person
person)
#2: (person_organization_select = organization
organization)
#3: (person_organization_select = person_and_organization
person_and_organization)

5.1.4.2 DRAWING

AIM element: draughting_drawing_revision
Source: ISO 10303-201
Reference path: draughting_drawing_revision <=
drawing_revision

5.1.4.2.1 drawing_number

AIM element: drawing_definition.drawing_number
Source: ISO 10303-101
Reference path: draughting_drawing_revision <=
drawing_revision
drawing_revision.drawing_identifier ->
drawing_definition
drawing_definition.drawing_number

5.1.4.2.2 drawing_revision_id

AIM element: drawing_revision.revision_identifier
Source: ISO 10303-101
Reference path: draughting_drawing_revision <=
drawing_revision
drawing_revision.revision_identifier

5.1.4.2.3 drawing to approval

AIM element: PATH
Reference path: draughting_drawing_revision <=
drawing_revision
approved_item = drawing_revision
approved_item <-
draughting_approval_assignment.approved_items[i]
draughting_approval_assignment <=
approval_assignment

approval_assignment.assigned_approval ->
approval

5.1.4.2.4 drawing to drawing_sheet

AIM element: PATH
Reference path: draughting_drawing_revision <=
drawing_revision <=
presentation_set <-
area_in_set.in_set
area_in_set
{area_in_set =>
drawing_sheet_revision_usage}
area_in_set.area ->
presentation_area =>
drawing_sheet_revision

5.1.4.3 DRAWING_SHEET

AIM element: drawing_sheet_revision
Source: ISO 10303-101

5.1.4.3.1 sheet_number

AIM element: drawing_sheet_revision_usage.sheet_number
Source: ISO 10303-101
Reference path: drawing_sheet_revision <=
presentation_area <-
area_in_set.area
area_in_set =>
drawing_sheet_revision_usage
drawing_sheet_revision_usage.sheet_number

5.1.4.3.2 sheet_revision_id

AIM element: drawing_sheet_revision.revision_identifier
Source: ISO 10303-101

5.1.4.3.3 drawing_sheet to approval

AIM element: PATH
Reference path: drawing_sheet_revision
approved_item = drawing_sheet_revision
approved_item <-
draughting_approval_assignment.approved_items[i]

draughting_approval_assignment <=
approval_assignment
approval_assignment.assigned_approval ->
approval

5.1.4.4 ORGANIZATION

AIM element: (person)
(organization)
(person_and_organization)
Source: ISO 10303-41
ISO 10303-41
ISO 10303-41

5.1.4.4.1 organization_name

#1: If the organization is only a person
#2: If the organization is only an organization
#3: If the organization is a person within an organization

AIM element: #1: (person.id)
#2:(organization.name)
#3: ([person.id]
[organization.name])
Source: ISO 10303-41
ISO 10303-41
ISO 10303-41
ISO 10303-41
Reference path: #3: (person_and_organization
[person_and_organization.the_person ->
person
person.id]
[person_and_organization.the_organization ->
organization
organization.name])

Annex C

Revision history

This edition may be immediately implemented by any project interested in doing so. The use of this edition is mandated only for projects approved after the approval of this document by SC4. Projects that have not yet produced a Committee Draft at the time of this document's approval by SC4 are strongly encouraged to use this edition.

Projects using this edition of the document must use this edition alone, and in its entirety.

Within this edition, the use of templates is optional.

Bibliography

[1] ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

[2] ISO 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support.*

[3] ISO 10303-101:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 101: Integrated application resources: Draughting.*

[4] ISO 10303-201:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 201: Application protocol: Explicit draughting.*

[5] ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Application protocol: Associative draughting.*

[6] *ISO TC 184/SC4 organization handbook*, ISO TC 184/SC4 N492, October 1996.

[7] ISO TC 184/SC4/QC N135 *Compilation of issues and proposals for Guidelines for the development of mapping tables, SC4 N533*